CLAIMS:

What is claimed is:

1. A method of determining an edge for inversion in a cyclic compound directed graph, comprising:

evaluating a plurality of graph nodes of a graph for generation of a first node order subset pair;

determining the graph requires recursive evaluation for completing node ordering of the plurality of graph nodes;

dividing the graph into a plurality of graph partitions; and

generating respective node order subset pairs of the graph partitions.

2. The method of claim 1, wherein the step of evaluating further includes:

identifying a removable sink node of the graph;

removing the removable sink node from the graph; and

appending a designation associated with the removable sink node in a first node order subset of the first node order subset pair.

3. The method of claim 2, wherein the steps of identifying and removing are repeated until no removable sink nodes remain in the graph.

4. The method of claim 2, wherein the step of evaluating further includes:

identifying a removable source node of the graph;

removing the removable source node from the graph; and

appending a designation associated with the removable source node in a second node order subset of the first node order subset pair.

5. The method of claim 4, wherein the steps of identifying and removing the removable source node are repeated until no removable source nodes remain in the graph.

6. The method of claim 4, wherein the step of determining further includes:

identifying a plurality of top-level subgraph nodes in graph elements remaining after removal of the removable sink node and the removable source node.

7. A computer program product in a computer readable medium for identifying a graph edge for inversion in a cyclic compound directed graph, the computer program product comprising:

first instructions for receiving an input graph;

second instructions for initializing a first subset pair and iteratively identifying and removing removable sink nodes and removable source nodes from the input graph, the removable sink nodes recorded in a first subset of the first subset pair and the removable source nodes recorded in a second subset of the first subset pair;

third instructions, responsive to removal of the removable sink nodes and removable source nodes by the second instructions, that identify remaining graph elements; and

fourth instructions, responsive to the third instructions identifying the remaining graph elements, adapted to divide the remaining graph elements into graph partitions, wherein the second instructions initialize and record removable sink nodes and removable source nodes in respective subset pairs for each of the graph partitions.

8. The computer program product of claim 7, wherein the remaining graph elements include at least two top-level subgraph nodes.

9. The computer program product of claim 7, further including:

fifth instructions that concatenate respective subsets of each subset pair.

10. The computer program product of claim 9, wherein results of the concatenation of subset pairs associated with the partitions are appended to the second subset.

11. The computer program product of claim 10, wherein the first subset and the second subset are concatenated after appendage of the concatenation results.

12. The computer program product of claim 7, further including:

fifth instructions for generating a node order set including each node of the graph, wherein the node order set is generated from the first subset pair and the subset pairs associated with the partitions.

13. The computer program product of claim 12, further including:

sixth instructions for associating a respective sequence number with each of the nodes of the node order set.

14. The computer program product of claim 13, wherein the input graph is a data structure defining a plurality of graph edges each comprising a respective source node and a target node, the computer program product further including:

seventh instructions for comparing a sequence number of a source node of a first edge defined in a record of the data structure with a sequence number of a target node of the first edge, the seventh instructions, responsive to determining that the sequence number of the source node of the first edge is larger than the sequence number of the target node of the first edge, identify the first edge as an inversion edge.

15. The computer program product of claim 14, wherein the seventh instructions, responsive to identifying the first edge as an inversion edge, insert an asserted inversion value into the record.

16. The computer program product of claim 13, wherein the input graph is a data structure defining a plurality of graph edges each comprising a respective source node and a target node, the computer program product further including:

seventh instructions for comparing a sequence number of a source node of a first edge defined in a record of the data structure with a sequence number of a target node of the first edge, the seventh instructions, responsive to determining that the sequence number of the source node of the first edge is less than the sequence number of the target node of the first edge, identify the first edge as a non-inverted edge.

17. The computer program product of claim 16, wherein the seventh instructions, responsive to identifying the first edge as a non-inverted edge, insert a non-asserted inversion value into the record.

18. A data processing system for identifying a graph edge for inversion, comprising:

a memory that contains a set of instructions and a data structure defining a plurality of edges of a graph; and

a processing unit, responsive to execution of the set of instructions, for reading the data structure and generating a first subset pair having respective subsets of ordered nodes of the graph, the processing unit evaluating graph elements remaining after generation of the first subset pair and recursively generating a plurality of subset pairs responsive to identification of a plurality of graph elements remaining after generation of the first subset pair.

19. The data processing system of claim 18, wherein the remaining graph elements comprise a plurality of top-level subgraph nodes.

20. The data processing system of claim 18, wherein the processing unit concatenates the plurality of subset pairs and appends the concatenated subset pairs to a first subset of the first subset pair, execution of the set of instructions providing an identification of an edge of the plurality of edges for inversion.